4. b) O alta echipa de programatori a dezvoltat algoritmul de lock prezentat in pseudocodul urmator ce incapsuleaza un alt lock oarecare. Se considera ca lock-ul incapsulat asigura corect excluderea mutuala si este starvation-free. De asemenea lock-ul incapsulat permite un apel unlock fara exceptie si fara efect chiar daca nu a existat un apel lock. ThreadId se considera a fi o clasa ce furnizeaza un id unic pozitiv fiecarui thread.

Intr-o executie concurenta a n > 1 thread-uri, asigura acest algoritm excluderea mutuala? Argumentati.

```
1 class VeryShadyLock {
2    private Lock lock;
3    private volatile int x, y = 0;
4
5    public void lock() {
6       int me = ThreadId.get();
7       x = me;
8       while (y != 0) {};
9       y = me;
10      if (x != me) {
11         lock.lock();
12      }
13   }
14
15   public void unlock() {
16      y = 0;
17      lock.unlock();
18   }
19 }
```

Excluderea mutuala asigura ca, in orice moment, doar un singur thread poate accesa sectiunea critica.

In clasa VeryShadyLock, variabila volatila x (vizibila tuturor threadurilor) este utilizata pentru a indica threadul care doreste acces la sectiunea critica. De asemenea, variabila volatila y serveste ca indicator al starii sectiunii critice: cand y = 0, sectiunea critica este libera; cand y ≠ 0,

sectiunea critica este ocupata, iar valoarea lui y reprezinta ID-ul threadului care are in acel moment acces la sectiunea critica.

Algoritmul asigura excluderea mutuala, deoarece toate threadurile initial asteapta eliberarea sectiunii critice (prin bucla *while (y != 0) {};*). Dupa ce sectiunea critica este libera *(y = 0*), threadul curent seteaza *y = me*, semnaland ca doreste acces la sectiunea critica.

Pentru a se asigura ca doar un singur thread acceseaza sectiunea critica, se face o verificare suplimentara: *if (x != me)*. Aceasta verificare detecteaza cazul in care un alt thread a modificat intre timp valoarea lui x, semnaland ca si el doreste acces. In aceasta situatie, threadul curent va apela functia *lock.lock()*, utilizand mecanismul de blocare intern pentru a preveni accesul simultan la sectiunea critica.

Astfel, algoritmul garanteaza ca, la orice moment, doar un singur thread poate intra in sectiunea critica.